

WE CLAIM:

1. A programmable object model for an Extensible Markup Language (XML) capable application, comprising:
 - an application programming interface for allowing a user to
 - 5 programmatically access the functionality of a software application capable of applying XML functionality to a document;
 - the application programming interface comprising a message call for inserting an XML element into a location within the document; and
 - the application programming interface operative to receive a return value
 - 10 from the application responsive inserting the XML element into the location within the document.
2. A programmable object model for an Extensible Markup Language (XML) capable application, comprising:
 - an application programming interface for allowing a user to
 - 15 programmatically access the functionality of a software application capable of applying XML functionality to a document;
 - the application programming interface comprising a message call for accessing data contained within an XML element applied to the document; and
 - the application programming interface operative to receive a return value
 - 20 from the application, the return value associated with providing access to the data contained within the XML element applied to the document.
3. A programmable object model for an Extensible Markup Language (XML) capable application, comprising:
 - an application programming interface for allowing a user to
 - 25 programmatically access the functionality of a software application capable of applying XML functionality to a document;

the application programming interface comprising a message call for locating an XML element applied to the document, where an XPath query is passed as a parameter of the message call; and

the application programming interface operative to receive a return value
5 from the application, the return value associated with providing a location in the document of the XML element applied to the document.

4. A programmable object model for an Extensible Markup Language (XML) capable application, comprising:

an application programming interface for allowing a user to
10 programmatically access the functionality of a software application capable of applying XML functionality to a document;

the application programming interface comprising a message call for accessing one or more properties associated with one or more XML elements applied to the document; and

15 the application programming interface operative to receive a return value from the application, the return value associated with providing the one or more properties associated with one or more XML elements applied to the document.

5. A method for programmatically accessing the functionality of an
20 Extensible Markup Language (XML) capable software application, comprising:

calling the software application via an object-oriented message call;

passing an object property to the software application, the object property being associated with XML functionality of the software application; and

25 in response to the message call and the object property passed to the software application, receiving access to the XML functionality of the software application associated with the object property passed to the software application.

6. The method of claim 5, whereby passing the object property to the software application includes passing an object method to the software application for

transforming an XML document, the method object including as a parameter a path to an XSLT transformation file for transforming the XML document according to the XSLT transformation file.

5 7. The method of claim 6, whereby the method object passed to the software application further includes a parameter indicating whether the XSLT transformation file is to be applied to all data contained in the XML file or whether the XSLT transformation file is to be applied to only non-native XML markup data applied to the XML document.

10

8. The method of claim 5, whereby passing the object property to the software application includes passing an object property to the software application for controlling the appearance of an associated schema file text when an XML element associated with the schema file text is to be presented in the software application's user
15 interface.

9. The method of claim 5, whereby passing the object property to the software application includes passing an object property for controlling how the software application saves XML markup applied to a document.

20

10. The method of claim 5, whereby passing the object property for controlling how the software application saves the XML markup applied to the document includes passing an object property for causing the software application to save the XML markup applied to the document according to the native XML
25 functionality of the software application without application of any XSLT transformation prior to saving the XML markup applied to the document.

11. The method of claim 5, whereby passing the object property to the software application includes passing an object property to the software application for

causing the application to save only XML markup and associated data applied to the document.

12. The method of claim 5, whereby passing the object property to the software application includes passing an object property to the software application for causing the software application to apply and XSLT transformation to the XML markup and associated data applied to the document and any non-XML data contained in the document prior to saving the document.

13. The method of claim 5, whereby passing the object property to the software application includes passing the object property for causing the software application to apply the XSLT transformation to only the non-XML data contained in the document prior to saving the XML document.

14. The method of claim 5, whereby passing an object property to the software application includes passing an object property for controlling the location of an XSLT transformation to be applied by the software application to a document upon saving the document such that only the output of the XSLT transformation is saved.

15. The method of claim 5, whereby passing an object property to the software application includes passing an object property for controlling whether an XSLT transformation should automatically be applied to a document before the document is saved, such that only the results of the transformation are saved.

16. The method of claim 5, whereby passing an object property to the software application includes passing an object property for controlling whether XML markup applied to a document is displayed to the user along with text being edited by the user in a document.

17. The method of claim 5, whereby passing an object property to the software application includes passing an object property for controlling an appearance

of an XML element name as a visible placeholder in a document where no data has been entered for the XML element.

18. The method of claim 5, whereby passing an object property to the software application includes passing a method property for inserting XML markup into
5 a document at a specified location, whereby a text string associated with the XML markup to be inserted and whereby any XSLT transformations to be applied to the inserted markup are passed as parameters to the method property.

19. The method of claim 5, whereby passing an object property to the software application includes passing a method object to the software application for
10 creating an XML nodes collection object and for adding to the XML nodes collection object any new XML node objects, whereby a name for a new XML node object, a uniform resource identifier identifying a Namespace associated with a new XML node object, and a range pointer to a location in a document where the new XML node object is to be applied are passed to the software application as parameters of the method
15 object.

20. The method of claim 19, whereby passing an object property to the software application includes passing a method object for accessing individual XML node objects contained in the collection of XML node objects, whereby an identification representing the position of a requested XML node object in a Namespace
20 library is passed as a parameter of the method object.

21. The method of claim 5, whereby passing an object property to the software application includes passing an object property for returning a base name of a specified XML elements.

22. The method of claim 5, whereby passing an object property to the
25 software application includes passing an object property for pointing to an XML nodes

collection that consists of all XML elements that are child elements for a specified element.

23. The method of claim 5, whereby passing an object property to the software application includes passing a method property to the software application for
5 copying a specified XML element and all data associated with the XML element for pasting to a separate location within a document.

24. The method of claim 5, whereby passing an object property to the software application includes passing an object property for copying an XML element and all data associated with the XML element and for removing the copied XML
10 element and the copied data associated with the XML element from a document from which the XML element and the data associated with the XML element are copied.

25. The method of claim 5, whereby passing an object property to the software application includes passing a method property for removing a specified XML element from a document without affecting data associated with the XML element.

15 26. The method of claim 5, whereby passing an object property to the software application includes passing an object property for pointing to a first XML element that is a child element of a specified XML element.

27. The method of claim 5, whereby passing an object property to the software application includes passing an object property for indicating whether an XML
20 element has any child elements that contain no associated data.

28. The method of claim 5, whereby passing an object property to the software application includes passing an object property for pointing to a last XML element that is a child element of a specified XML element.

29. The method of claim 5, whereby passing an object property to the software application includes passing an object property for indicating whether an XML element is at an inline level, a paragraph level, a table cell level, a table row level, a table level, or other levels supported by the software application.

5 30. The method of claim 5, whereby passing an object property to the software application includes passing an object property for returning the uniform resource identifier of an XML schema file associated with a specified XML element.

31. The method of claim 5, whereby passing an object property to the software application includes passing an object property for returning a pointer to an
10 XML node that represents a next XML element immediately following a specified XML element.

32. The method of claim 5, whereby passing an object property to the software application includes passing an object property for indicating a type of an XML element, whereby the type of the XML element may include an XML element or
15 an attribute of an XML element.

33. The method of claim 5, whereby passing an object property to the software application includes passing an object property for returning a pointer to a document containing a specified XML element.

34. The method of claim 5, whereby passing an object property to the
20 software application includes passing an object property for returning a pointer to an XML element that is a parent XML element of a specified XML element.

35. The method of claim 5, whereby passing an object property to the software application includes passing an object property for controlling whether a placeholder text is displayed in place of XML elements applied to an XML element
25 applied to a document when the XML element contains no associated data.

36. The method of claim 5, whereby passing an object property to the software application includes passing an object property for returning a pointer to an XML element that is a previous XML element before a specified XML element.

37. The method of claim 5, whereby passing an object property to the software application includes passing an object property for returning a pointer to a range in a document contained by a specified XML element.

38. The method of claim 5, whereby passing an object property to the software application includes passing a method property for removing an XML element that is a child XML element of a specified XML element, whereby a pointer to the child XML element is passed with the method object as a parameter.

39. The method of claim 5, whereby passing an object property to the software application includes passing a method property for finding all XML elements in a document that match a specified XPath query whereby an XML elements collection consisting of all XML elements matching the specified XPath query are returned.

40. The method of claim 5, whereby passing an object property to the software application includes passing an object property for returning as plain text any data entered into a document associated with a specified XML element.

41. The method of claim 5, whereby passing an object property to the software application includes passing an object property for returning an XML markup representation of a specified XML element and all data associated with the specified element.